COMMON MEMORY 27

	A	B	C	D	E	- - - -
a	01	02	03	04	05	
b	Zα1	08	Zα2	Zα3	09	
c	Zβ1	010	07	Zδ1	011	
d	Zβ2	Zδ3	Zδ4	Zδ5	Zδ7	
e	Zδ2	06	012	Zμ1	Zδ6	
.						
.						
.						

FIG. 5

which is determined on which outgoing lines **12** the respective cell Z should be sent, and the new header for the transmitted cell. This action is shown by the dotted arrows **21**. Cells Z which contain no user information, (so-called empty cells) are removed from the buffers **15**, as shown by the arrows **22**. On the other hand, when there are no cells Z to send out on the outgoing lines **12**, empty cells are transmitted instead, as shown by the arrows **23**. Cells can be sent out in a multicast manner or as a general broadcast, and also on several pre-determined lines or on all of them.

The ATM-node **10** performs the functions of a switching exchange for the continuous onward routing of incoming cells Z to outgoing lines **12**, in accordance with the description. Of course, in this respect, consideration must be given to the associated requirement that, whenever possible, no cell-loss should occur.

FIG. **2** shows the block diagram of one such ATM-node **10**, which has a common memory **27**, with a third plurality k of storage blocks. Basically, every arriving cell Z is stored in this common memory **27**, before being sent out again on one of the outgoing lines **12**. By means of its extensive buffering capability, this common memory **27** prevents cell-loss and is preferably arranged so that, in a sequential manner, only one single cell Z can be written in or read out, respectively, at a time. ie. it operates serially. To achieve this, a multiplexer **29** is inserted between the buffers **15** on the incoming-side and the common memory **27**. This takes data from the input buffers **15** in a cyclic order.

In a similar manner as at the incoming-side of the ATM-node **10**, a demultiplexer **30** is inserted between the common memory **27** and the buffers **16** for the outgoing lines **12**. This demultiplexer routes cells Z read out of the common memory **27**, to the respective outgoing lines **12**, as had been determined at the incoming-side from the reading of the header information by the address-reading unit **33**.

The common memory **27** is controlled by a control unit **35** which receives the necessary control information exclusively from the address-reading unit **33**. Finally a sequencer **38** serves to co-ordinate the units of the ATM-node **10** with the common memory **27**. The sequencer triggers the individual function steps of the various units and synchronises them to each other.

FIG. **3** shows the block diagram of the control unit **35** and the sequencer **38**, which were shown in FIG. **2**. The most important units of the control unit **35** are three addressable read/write memories **41**, **43**, **45**, (preferably RAMs—random access memories). Each has an index address input line Adr and an input/output line Data for storing data into the storage locations.

The first read/write memory **41** is called hereafter the H-store (H=head). It contains m+1 storage locations, where m is the term for the second plurality of outgoing lines **12**. (In the following example m=4). The storage locations of the H-store are individually associated, each with one of these outgoing lines **12**, and denoted by the greek letters α β,∂,μ. In addition, there is a (m+1)th location denoted by O, which is associated with a list of the empty locations (as described later).

The content of the storage locations O, α,β,∂,μ of the H-store **41** is respectively an address of a header element (as described later).

The H-store **41** is further regulated by the sequencer **38** and a pointer controller depending upon the incoming cells, whereby the contents of the storage locations α,β, ∂,μ are respectively written in or read out (as described later). The storage location O receives, respectively, special treatment.

The second read/write memory **43** is called hereafter the T-store (T=tail). It is constructed in a similar way to the H-store. Its storage locations are likewise denoted in accordance with the greek letters or, α,β,∂,μ, together with **0**. The content of the storage locations is also, respectively, an address. However, the pointer controller selects the storage locations in a cyclic manner.

The third read/write memory **45** is called hereafter the P-store (P=pointer). It builds an image of the common memory **27**. It contains the same number k of storage locations as there are storage blocks in the common memory **27**. The storage locations and the storage blocks have, pair-wise, the same address. The difference between the two stores is that a content of the storage locations of the P-store is a (short) address, whereas the respective content of the storage blocks of the common memory **27** is a (long) cell Z.

In addition, the control unit **35** contains a control information input **46**, input logic **48**, bus- and control- logic **50**, an output **52**, a comparator **59**, monitor logic **71** and priority logic **73**. All these units are switched on, or triggered, sequentially, by means of the sequencer **38**, as indicated by the angled arrows.

The input logic **48** makes the connection between the control information input **46** (coming from the address reading unit **33**, FIG. **2**), the monitor logic **71** as well as the priority logic **73**, and the address input lines Adr of the H-store **41** and the T-store **43**.

FIG. **4** shows the bus- and control- logic **50** in more detail. The following discussion makes reference to items depicted in FIGS. **3** and **4**. This logic **50** consists of a number of co-ordinated buses **66**, **67**, **68**, **69** and of control elements such as multiplexers **51** and transceivers **53**, **55**, **57** inserted inbetween. The logic **50** connects the input and output Data lines of the H-store **41** and the T-store **43** with both input lines (Adr and Data) of the P-store **45**, with the monitor logic **71**, with the comparator **59** and with the output **52**. The path to the output **52** is via two more output buffers **61**, **63**. The output **52** itself is a uni-directional bus which goes to the common memory **27**, where it is connected to the address input line. The input logic **48** consists essentially of two signal multiplexers **47**, **49** which co-ordinate respectively the address signals for the H-store **41** and the T-store **43**. The inputs shown as **0** on the multiplexers **47**, **49** are for selecting the list of empty locations.

The priority logic **73**, which is described more thoroughly later, is an option for the control unit **35**. It only has any relevance when cells Z are associated with different priorities for being forwarded, e.g. "urgent" and "normal". At first, however, only cells with a single priority are considered.

The way in which the common memory **27**—and the appertaining control unit **35**—operate is described by means of an example in the next figures, which is associated with the state of the stores at particular instances. In addition, FIG. **5** shows the addresses and the current content of the storage blocks of the common memory **27**. As an example, is memory can be constructed as a grid, whereby the columns are associated with the addresses A, B, C, D, E, . . . and the rows are associated with the addresses a, b, c, d, e, . . . For practical reasons, only the content of the twenty-five storage blocks with block addresses Aa to Ee are shown.

The content of the storage blocks of the common memory **27** form—as described—intermediately stored cells Z, or else the content is empty. In the latter case, the block is marked with a O plus a sequence number **1**, **2**, **3**, . . . In the other case, ie. when a cell Z is stored, the block is marked